

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Nejc Poljanšek

**Vzpostavitev spletnega vmesnika za prikaz  
tenziomiografskih meritev**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2014



UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Nejc Poljanšek

**Vzpostavitev spletnega vmesnika za prikaz  
tenziomiografskih meritev**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Miha Moškon

Ljubljana, 2014



Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo: Vzpostavitev spletnega vmesnika za prikaz tenziomiografskih meritev.

Tematika naloge:

Kandidat naj v diplomskem delu vzpostavi spletni vmesnik, ki bo dostopal do tenziomiografskih (TMG) meritev shranjenih v oblaku in omogočal njihov prikaz in analizo. Še posebej naj se osredotoči na t. i. trend report, ki omogoča spremljanje posameznikovega napredka skozi čas. Vmesnik naj bo uporabniku prijazen in primeren za ciljno populacijo, tj. trenerje poklicnih športnikov.



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Nejc Poljanšek, z vpisno številko **63040291**, sem avtor diplomskega dela z naslovom:

*Vzpostavitev spletnega vmesnika za prikaz tenziomiografskih meritev*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Mihe Moškona,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 25. septembra 2014

Podpis avtorja:





## **Zahvala**

*Posebna zahvala gre mentorju doc. dr. Mihi Moškonu za pomoč, veliko mero potrpežljivosti in koristne nasvete pri pisanju diplomske naloge.*

*Zahvaljujem se tudi svojim staršem, ki so me spodbujali ter moralno in finančno podpirali vsa leta študija.*



# Kazalo

<b>Poglavje 1</b>	<b>Uvod .....</b>	<b>1</b>
<b>Poglavje 2</b>	<b>Tenziomiografija.....</b>	<b>3</b>
<b>Poglavje 3</b>	<b>Razvojna orodja, metode in tehnologije .....</b>	<b>5</b>
3.1	Uporabljen programski orodja.....	5
3.1.1	Navicat Premium .....	5
3.1.2	Tomcat Apache.....	5
3.1.3	Sublime text.....	6
3.1.4	Laravel framework .....	6
3.2	Uporabljene metode in tehnologije.....	8
3.2.1	MVC arhitektura.....	8
3.2.2	HTML.....	10
3.2.3	CSS .....	10
3.2.4	JavaScript .....	10
3.2.5	SQL.....	11
3.2.6	PHP .....	11
<b>Poglavje 4</b>	<b>Aplikacija .....</b>	<b>13</b>
4.1	Arhitektura sistema .....	13
4.2	Podatkovna baza .....	14
4.3	Dostopanje do podatkov in njihova interpretacija .....	16
4.4	Prikazovanje podatkov.....	16
4.5	Uporaba aplikacije .....	18
4.5.1	Prijavno okno.....	19
4.5.2	Izbira merjenja .....	19
4.5.3	Izbira funkcionalnosti .....	20
4.5.4	Prikaz spreminjanja karakterističnih točk skozi čas .....	21
4.5.5	Prikaz spreminjanja karakterističnih točk skozi čas .....	23
4.5.6	Pregled vrednosti posameznih meritev .....	24

<b>Poglavje 5</b>	<b>Zaključek .....</b>	<b>27</b>
-------------------	------------------------	-----------

## Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>PHP</b>	Personal Home Page Tools	Orodja za osebno spletno stran
<b>TMG</b>	Tensiomyography	Tenziomiografija
<b>MVC</b>	Model-View-Controller	Model-Pogled-Kontroler
<b>EKG</b>	Electrocardiography	Elektrokardiografija
<b>SQL</b>	Structured Query Language	Strukturirani povpraševalni jezik
<b>HTML</b>	Hyper Text Markup Language	Označevalni jezik za izdelavo spletnih strani
<b>CSS</b>	Cascading Style Sheets	Stilne predloge
<b>ORM</b>	Object-relational mapping	Objektno-relacijske preslikave
<b>CLI</b>	Command-line interface	Vmesnik z ukazno vrstico
<b>ASP</b>	Active Server Pages	Aktivne strežniške strani
<b>XML</b>	Extensible Markup Language	Razširljiv označevalni jezik
<b>XHTML</b>	Extensible HyperText Markup Language	Razširljiv označevalni jezik za izdelavo spletnih strani
<b>HTTP</b>	HyperText Transfer Protocol	Protokol za prenos spletnih strani
<b>TCP</b>	Transmission Control Protocol	Protokol za nadzor prenosa
<b>IP</b>	Internet protocol	Internetni protocol



## Povzetek

V sklopu diplomske naloge smo razvili rešitev za prikaz tenziomiografskih meritev. Spletna aplikacija omogoča uporabniku dostop do podatkov o meritvah, ki se nahajajo v podatkovni bazi v oblaku. Za razvoj spletnega vmesnika smo uporabili odprtokodno ogrodje Laravel, ki temelji na programskem jeziku PHP.

Podrobneje smo predstavili arhitekturo celotnega sistema ter uporabniški vmesnik spletne aplikacije. Predstavili smo tudi glavne funkcionalnosti aplikacije ter opisali način uporabe spletnega vmesnika. Pri tem smo pridobili uporabne izkušnje pri izgradnji spletne aplikacije, ki temelji na ogrodju Laravel.

**Ključne besede:** tenziomiografske meritve, spletna aplikacija, podatkovna baza, oblak, Laravel, PHP, uporabniški vmesnik





## **Abstract**

The purpose of this thesis was to develop a solution to display tensiomyographic measurements. The web application enables the user to access the measurement data stored in the cloud database. The open source Laravel framework written in the PHP programming language was used for developing the web interface.

The architecture of the whole system and the web app user interface are introduced in more details. The main app functionalities and the way of using the web interface are also described. While building the web application based on the Laravel framework, useful experiences were gained.

**Keywords:** tensiomyographic measurements, web application, database, cloud, Laravel, PHP, user interface



## **Poglavje 1      Uvod**

Tenziomiografija (TMG) je učinkovit in neinvaziven postopek za merjenje mišičnega odziva [1]. Na začetku se je metoda uporabljala predvsem v medicini, v zadnjem času pa se vse bolj uporablja tudi pri treningu profesionalnih športnikov [2]. S pomočjo rezultatov opravljenih meritev lahko trenerji določijo optimalen režim treningov, s čimer športniki dosegajo boljše rezultate in obenem zmanjšajo nevarnosti za nastanek poškodb.

V delu se osredotočimo na uporabo TMG metode za trenerje poklicnih športnikov. Podatki, ki jih z meritvami pridobimo, v grobi obliki ciljni populaciji ne povejo nič ali pa zelo malo. V ta namen je potrebno podatke obdelati in prikazati v ustrezni obliki (npr. z uporabo grafiknov).

V delu predstavimo razvoj vmesnika za obdelavo in prikazovanje zajetih meritev na skupini športnikov s poudarkom na napredovanju oziroma nazadovanju posameznih mišic skozi čas (t. i. trend report). Za predstavitev vrednosti meritev uporabimo linearne grafikone, ki najbolj nazorno prikazujejo spreminjanje vrednosti skozi čas. Skozi razvoj spletnega vmesnika pridobimo znanje o uporabi MVC arhitekture.

V prvem delu diplomskega dela je predstavljena TMG metoda, njene karakteristične točke, ki jih pridobimo s posamezno meritvijo. V drugem delu predstavimo orodja in metode za razvoj spletnega vmesnika. V tretjem delu je opisana arhitektura sistema, dostopanje in interpretacija podatkov ter podrobnejši opis razvite aplikacije.



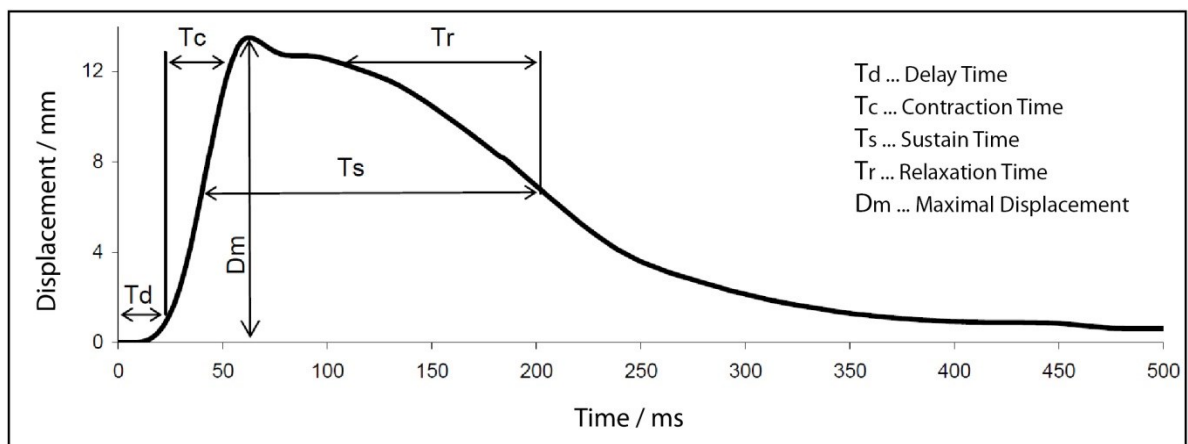
## Poglavje 2     Tenziomiografija

Tenziomiografija (TMG) je metoda, ki meri mišično aktivnost in je edinstveno orodje za zgodnjo diagnostiko in odkrivanje mišičnih nesimetrij [1]. Omogoča optimizacijo treningov, preprečevanje poškodb in pospeševanje rehabilitacijskih procesov. Podobno, kot predstavlja EKG za srce, predstavlja TMG za mišice [2]. Metoda je bila razvita že v poznih osemdesetih letih. V osnovi je bila metoda zasnovana za rabo v medicini, vendar so jo leta 1996 prvič uporabili tudi na področju športa. Danes je bila metoda uporabljena že na najboljših športnikih na svetu [3].

Do meritev pridemo s pomočjo posebnega senzorja, ki zaznava krčenje mišice. Senzor namestimo na mišico, ki jo želimo meriti, krčenje mišice pa umetno vzbudimo z elektronskim stimulatorjem. Senzor meritev zabeleži kot funkcijo odvisnosti od časa. Vsako krčenje mišice lahko prikažemo v obliki krivulje. Iz krivulje lahko razberemo pet različnih podatkov oziroma parametrov, ki jih uporabljamo za nadaljnje postopke (Slika 2.1).

Parametri, ki jih razberemo iz krivulje krčenja mišice so:

- čas zakasnitve (angl. *delay time*, krajše Td ), ki ga merimo v milisekundah in pomeni pretečen čas, ki ga mišica potrebuje, da se odzove na elektronsko vzbujeno krčenje;
- čas krčenja (angl. *contraction time*, krajše Tc), ki ga merimo v milisekundah in pomeni pretečen čas, ki ga mišica potrebuje, da se skrči;
- čas vzdrževanja (angl. *sustain time*, krajše Ts), ki ga merimo v milisekundah in predstavlja pretečen čas od trenutka, ko mišica pri krčenju preseže srednjo vrednost premika mišice, do trenutka, ko pade nazaj pod srednjo vrednost;
- čas sprostitve (angl. *relaxation time*, krajše Tr), ki ga merimo v milisekundah in predstavlja pretečen čas, ki ga mišica potrebuje, da se sprosti do stanja, ko pade pod srednjo vrednost premika mišice,
- maksimalen premik (angl. *maximal displacement*, krajše Dm), ki ga merimo v milimetrih in predstavlja maksimalen premik mišice pri krčenju.



Slika 2.1: Prikaz karakterističnih točk in njihova povezava.

## **Poglavje 3      Razvojna orodja, metode in tehnologije**

Danes za razvoj aplikacij obstaja ogromno različnih razvojnih orodij. Katera razvojna orodja uporabimo pri razvoju aplikacije, je odvisno od mnogo dejavnikov. Razlika med njimi se kaže v načinu uporabe, funkcionalnostih, katere ponuja, ceni, predvsem pa v uporabnosti. Učinkovit in hiter razvoj aplikacij je v prvi vrsti odvisen od znanja programiranja, zato je posledično pomembna tudi izbira najprimernejših razvojnih orodij. Za izbiro najprimernejšega pa moramo tudi vedeti, kaj nam določeno razvojno orodje ponuja.

V nadaljevanju si bomo najprej pogledali pregled uporabljenih programskih orodij, ki mu bo sledil opis uporabljenih metod in tehnologij.

### **3.1      Uporabljena programska orodja**

Pogledali si bomo, katera razvojna in ostala programska orodja so bila uporabljena pri razvoju spletne aplikacije.

#### **3.1.1 Navicat Premium**

Navicat premium je grafično orodje za administracijo in razvoj podatkovnih baz. Z njim lahko urejamo in vzdržujemo različne podatkovne baze, poganjamo SQL poizvedbe in skripte ter urejamo in razhroščujemo SQL stavke [4]. Vsebuje vrsto funkcionalnosti za delo s podatkovnimi bazami ter ostaja hiter in enostaven za uporabo.

#### **3.1.2 Tomcat Apache**

Tomcat Apache je spletni strežnik, ki igra ključno vlogo pri širjenju spleta. Bil je prva alternativa Netscapeovemu spletnemu strežniku, trenutno znanemu kot spletni strežnik Sun Java System. Od aprila 1996 je Apache najbolj popularen HTTP strežnik na celotnem spletu, od oktobra 2007 pa je bilo na Apachejevih strežnikih postavljenih približno 48 % vseh spletnih strani [5].

### 3.1.3 Sublime text

Spletna aplikacija je napisana v urejevalniku izvirne kode Sublime Text. To je tekstovni urejevalnik za urejanje vseh vrst programskih sintaks [6]. Gre za zelo preprost urejevalnik, ki je zelo pregleden in hiter, kljub temu pa omogoča samodopolnjevanje kode, poskrbi za samodejno zamikanje kode, označuje sintaktične napake ter ponuja še ogromno drugih funkcij za enostavno delo z besedilom in različnimi programskimi sintaksami. Omogoča kreacijo projektov, v katerih lahko uporabljamo različne metode po naših željah in zahtevah. Pri razvoju spletne aplikacije na določenem PHP ogrodju je to zelo učinkovito, saj moramo že od samega začetka skrbeti za pravilno urejeno direktorijsko strukturo aplikacije. Istočasno uporabljamo različne sintakse oziroma metode. V enem urejevalniku hitro in pregledno urejamo vse od osnovne HTML kode, PHP skript in JavaScript-a, Laravel sintakse, pa do SQL poizvedb do podatkovne baze in urejanja kaskadnih stilskih predlog (CSS).

### 3.1.4 Laravel framework

Laravel framework je brezplačno odprtokodno ogrodje, ki temelji na jeziku PHP. Namenjeno je razvoju spletnih aplikacij z MVC arhitekturo, ki jo bomo podrobneje opisali kasneje. V letu 2013 je bilo ogrodje Laravel navedeno kot najpopularnejše PHP ogrodje. Glede na ostala PHP razvojna ogrodja je Laravel manjše [7]. Ogrodje ima ogromno funkcionalnosti in različnih knjižnic, katere omogočajo izgradnjo tudi najzahtevnejših spletnih aplikacij. Spletna aplikacija je razvita na verziji ogrodja 4.2. V nadaljevanju so opisane glavne funkcionalnosti ogrodja Laravel.

#### 3.1.4.1 Blade

Blade je preprost, ampak zelo močan sistem za delo s predlogami. Deluje na podlagi dedovanja predlog in manjših sekcij. Omogoča preprosto sestavljanje pogledov. Vse Blade predloge imajo končnico `.blade.php`. Slika 3.1 prikazuje preprosto predlogo z uporabo sistema Blade. Uporabljeni sta dve različni vrsti sekcij. Prva, z imenom navigacija, bo vidna tudi na vseh podstraneh. Druga sekcija, z imenom vsebina, pa predstavlja območje, v katero se nalagajo vsebine ostalih delnih strani (angl. *partials*).



```

1  <html>
2  <body>
3    <nav class="navigation">
4      @section('navigacija')
5        <a href="/">Domov</a>
6        <a href="/about">Kontakt</a>
7      @show
8    </nav>
9    <div class="container">
10     @yield('vsebina')
11   </div>
12 </body>
13 </html>

```

Slika 3.1: Preprosta predloga z uporabo sistema Blade.

Blade vsebuje tudi druge sintaktične strukture, ki nam omogočajo lažje, preglednejše in hitrejše pisanje PHP sintakse. Slika 3.2: Običajna PHP sintaksa. prikazuje običajno PHP sintakso za nekaj preprostih ukazov, Slika 3.3 pa Blade sintakso, ki je preglednejša in vrne identični rezultat.

```

3  <?php if ($user->isLogged()): ?>
4    Dobrodošli nazaj, <strong><?php echo $user->name; ?></strong>
5  <?php endif; ?>

```

Slika 3.2: Običajna PHP sintaksa.

```

9  @if ($user->isLogged())
10    Dobrodošli nazaj, <strong>{{ $user->name }}</strong>
11 @endif

```

Slika 3.3: Blade sintaksa.

#### 3.1.4.2 Eloquent

Eloquent je ORM sistem, ki omogoča preprosto delo s podatkovno bazo na objektnem nivoju. Model entitete se v projektu vedno nahaja v direktoriju `app/models`. Vsak model razširja razred Eloquent.

#### 3.1.4.3 Migrations

Migracije (angl. *migrations*) predstavljajo izviren sistem kreiranja in verzioniranja podatkovne baze. Omogočajo spreminjanje sheme podatkovne baze ter ostati na tekočem o trenutnem stanju sheme.

#### **3.1.4.4 Router**

Usmerjevalnik (angl. *router*) je ena izmed prvih datotek, ki se naloži v naši aplikaciji. V njej se nahajajo definicije naših URL poti.

#### **3.1.4.5 Artisan CLI**

Vmesnik Artisan je v obliki ukazne vrstice vključen v ogrodje Laravel. Ponuja vrsto koristnih ukazov za uporabo med razvijanjem aplikacije. Temelji na komponenti Symfony Console.

#### **3.1.4.6 Helpers**

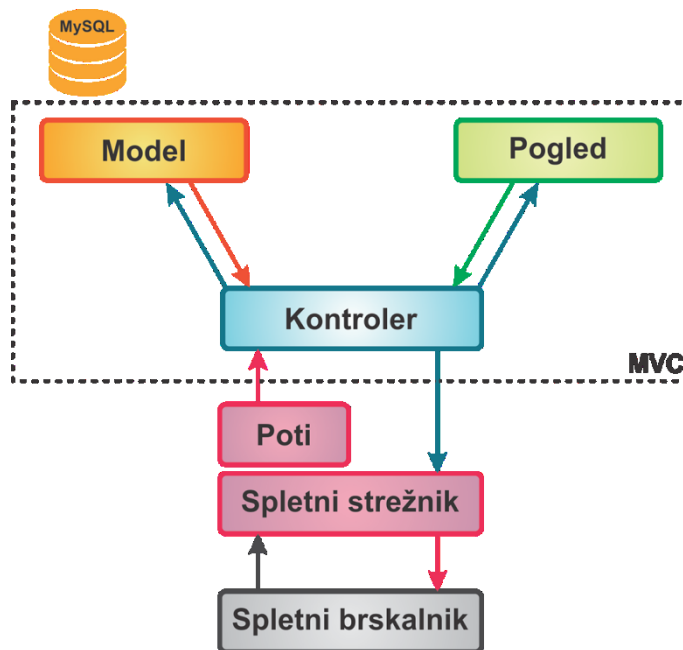
Objekti, ki pomagajo pri pogostih opravilih.

### **3.2 Uporabljene metode in tehnologije**

V tem poglavju so opisane metode in tehnologije, ki smo jih uporabili pri razvoju spletne aplikacije.

#### **3.2.1 MVC arhitektura**

MVC (angl. *Model-View-Controller*) omogoča razvoj modularnih rešitev [8]. Celoten vzorec v osnovi stremi k temu, da je potrebno ločiti tri različne vidike uporabniškega vmesnika: podatkovni model, predstavitveni model in vmesnik za komunikacijo med tema dvema modeloma – kontroler, kot prikazuje Slika 3.4. Primarna ideja pri zasnovi, realizaciji in vzdrževanju je, da vse tri komponente ločimo tako, da so popolnoma neodvisne ena od druge. S tem dosežemo to, da pri spreminjanju katerekoli komponente ne vplivamo na ostale [9]. Na primer uporabniški vmesnik lahko posodobimo s povsem novim videzom brez posegov na podatkovnem modelu ali kontrolerju. MVC arhitekturo uporablja večina ogrodij za spletno programiranje, od PHP ogrodij, pa vse do ASP.NET MVC projektov. Sicer pa gre za neverjetno pogost vzorec, ki ga pogosto srečujemo. Primer je lahko tudi večina spletnih strani, ki je narejena v HTML jeziku in uporablja CSS predloge. Tako imamo za model HTML datoteko, CSS predstavlja pogled, kontroler pa je sam spletni brskalnik.



Slika 3.4: "Model-View-Controller" arhitektura.

#### 3.2.1.1 Model

Model skrbi za komunikacijo s podatkovno bazo. Izvršuje povpraševanja in vrača rezultate nazaj. Poglede in kontrolerje, katerim pripada obveščanje, ko pride do sprememb njegovega stanja. Taka opozorila omogočajo pogledu, da posodobi izpis ter kontrolerju, da prilagodi vrsto ukazov, ki so na voljo. V nekaterih primerih, ko gre za pasivno implementacijo MVC, morajo komponente same povpraševati model po spremembah.

#### 3.2.1.2 Pogled (angl. View)

Predstavlja uporabniški vmesnik in je vizualna prezentacija svojega modela. Od modela dobi podatke s postavljanjem vprašanj, lahko pa model tudi posodobi s pošiljanjem primernih sporočil. Vsa ta vprašanja in sporočila morajo biti v terminologiji modela, tako da pogled pozna semantiko atributov modela, ki ga predstavlja. Končni rezultat pogleda je prikazan v obliki HTML strani.

#### 3.2.1.3 Kontroler (angl. Controller)

Vsa logika aplikacije poteka preko kontrolerjev. Procesira in manipulira zahteve in podatke. Le-ta povezuje model in pogled, prav tako pa povezuje uporabnika s sistemom. Uporabniku omogoča vnos podatkov s pripravo ustreznih pogledov na zaslon, vse podatke ustrezno zajame in jih pretvori v sporočila, ki jih posreduje naprej.

### 3.2.2 HTML

HTML (angl. *Hyper Text Markup Language*) je označevalni jezik za izdelavo spletnih strani. Predstavlja osnovo spletnega dokumenta. HTML dokument vsebuje zaporedje ukazov, ki določajo, kako naj se spletna stran prikaže [10]. Ukazom z drugo besedo rečemo tudi značke (angl. *tags*), ki morajo biti pravilno uporabljene. Običajno nastopajo v paru (npr. `<p>` in `</p>`) ter so med seboj ugnezdene. Značke so zapisane v špičastih oklepajih (npr. kot `<p>`). Vsak HTML dokument je sestavljen iz dveh glavnih delov. Prvi je glava (angl. *head*), v katerem se običajno vnese naslov strani (angl. *title*), sledi mu telo (angl. *body*), v katerem je zapisana vsa ostala vsebina dokumenta.

### 3.2.3 CSS

CSS je preprost slogovni jezik, ki skrbi za izgled spletnih strani. Z njim definiramo stil HTML elementov v obliki pravil za prikaz v spletnem brskalniku [11]. V pravilih lahko določamo sloge, velikosti, barve, poravnave, obrobo, stopnjo prosojnosti, odmike, pozicije in še ogromno drugih atributov, prav tako pa tudi aktivnosti, ki jih uporabnik spletne strani izvaja (npr. prekritje povezave z miško). CSS se je prvič pojavil leta 1996 in se je do danes že zelo izpopolnil. Z novimi verzijami lahko nadziramo vedno več elementov in jim nastavljamo vedno več lastnosti. Glavni namen CSS je, da vsebinski del spletne strani oziroma dokumenta loči od njene predstavitve. S tem omogočimo lažje urejanje in dodajanje stilov ter poskrbimo za večjo preglednost dokumentov temelječih na HTML sintaksi. Z uvedbo CSS se izognemo ponavljanju kode, zmanjša se kompleksnost dokumentov spletne strani in s tem poveča preglednost kode. Več strani lahko uporablja iste oblikovne predloge, brez da bi jih posebej pripisovali vsaki izmed njih. Obratno lahko eno stran prikažemo z različnimi stilskimi predlogami. Za delovanje CSS jezika je potreben spletni brskalnik, ki omogoča CSS podporo. Na žalost nekateri brskalniki CSS pravil ne obravnavajo identično ali pa se določenim pravilom izogibajo, zato lahko pride do različnih prikazov iste spletne strani v različnih spletnih brskalnikih. CSS pravila se običajno nahajajo v svoji datoteki (npr. `style.css`), lahko pa se nahajajo tudi v samem HTML dokumentu.

### 3.2.4 JavaScript

JavaScript je skriptni programski jezik, ki se uporablja pri izdelavi interaktivnih spletnih strani, ki se izvajajo na odjemalcu. Omogoča funkcionalnosti kot so meniji, zvoki in druge interaktivne funkcije, validacijo spletnih obrazcev, funkcije za komunikacijo s strežnikom itd. Spada v skupino tolmačev (angl. *interpreter*), kar pomeni, da za poganjanje skript ne potrebujemo prevajalnika, ki bi iz izvirne kode naredil programsko kodo, saj tolmač izvirno kodo programa tolmači sproti. Tolmač za JavaScript je vgrajen v vsak novejši spletni

brskalnik [12]. Skripte z izvorno kodo se lahko nahajajo v posebni datoteki (npr. program.js), lahko pa jih uporabljamo neposredno v dokumentu HTML, med značkama `<script>` in `</script>`. Veliko se uporablja tudi knjižnjice, kot je jQuery [13], ki vsebuje zbirko funkcij za hitrejši razvoj spletnih aplikacij. Rešuje tudi težave v zvezi s spletnimi brskalniki, kjer se isto stvar za različne brskalnike implementira drugače.

### 3.2.5 SQL

SQL (angl. *Structured Query Language*) je najbolj razširjen in standiziran jezik za delo s podatkovnimi bazami. Omogoča kreiranje, spreminjanje, brisanje in manipuliranje s podatki, ki so shranjeni v relacijski podatkovni bazi [14].

### 3.2.6 PHP

PHP (angl. *PHP Hypertext Preprocessor*) je zelo razširjen odprtokodni programski jezik, ki se uporablja za razvoj dinamičnih spletnih strani. Napisan je bil v jeziku C leta 1994. PHP koda se lahko enostavno kombinira s HTML kodo. PHP primarno teče na spletnem strežniku, kjer jemlje PHP izvorno kodo za vhod in generira spletno stran kot izhod. To pomeni, da na lokalnem računalniku brez strežniške opreme PHP skript ne moremo izvajati, lahko pa si v spletnem brskalniku ogledamo spletno stran, katero je generiral spletni strežnik s PHP podporo. Tak strežnik ima namreč pre-procesor hiperteksta, ki odpre skripto, z interpreterjem izvede ukaze v njej, zapiše njene morebitne izpise v HTML, XML ali XHTML dokument in ga pošlje našemu računalniku. Na našem računalniku ga obravnava spletni brskalnik in poskrbi, da se prikaže na zaslonu kot spletna stran. Uporabnik ne ve, da je spletna stran napisana v PHP jeziku, saj je izvorna koda strani, ki jo lahko prikaže brskalnik, običajna HTML koda [15].



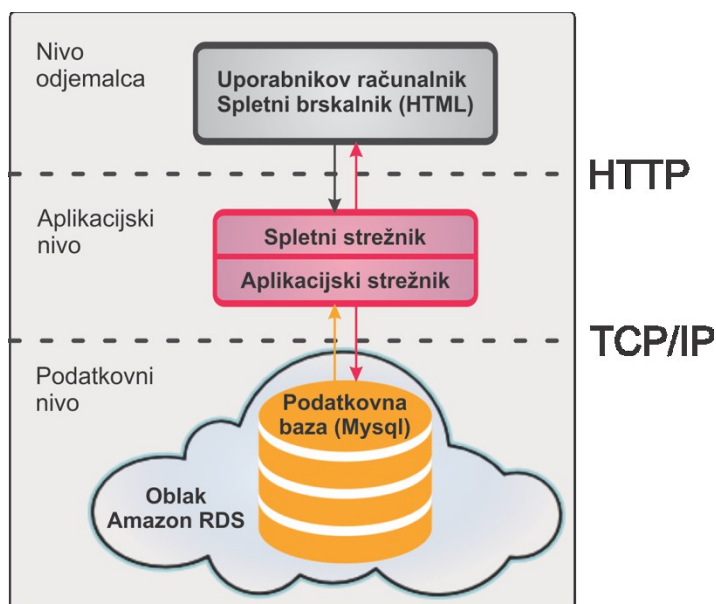
## Poglavje 4 Aplikacija

Glavna zahteva pri razvoju aplikacije je bila, da uporabniku omogoča pregled nad spreminjanjem meritev skozi čas. Aplikacija je bila razvita v obliki spletnega vmesnika, tako da je dostopna od povsod, kjer imamo dostop do interneta. Uporabniku omogoča pregled meritev posameznega merjenja ter prikaz spreminjanja karakterističnih točk skozi čas v obliki grafikonov.

V nadaljevanju sledi podrobnejši opis strukture in delovanja razvite aplikacije.

### 4.1 Arhitektura sistema

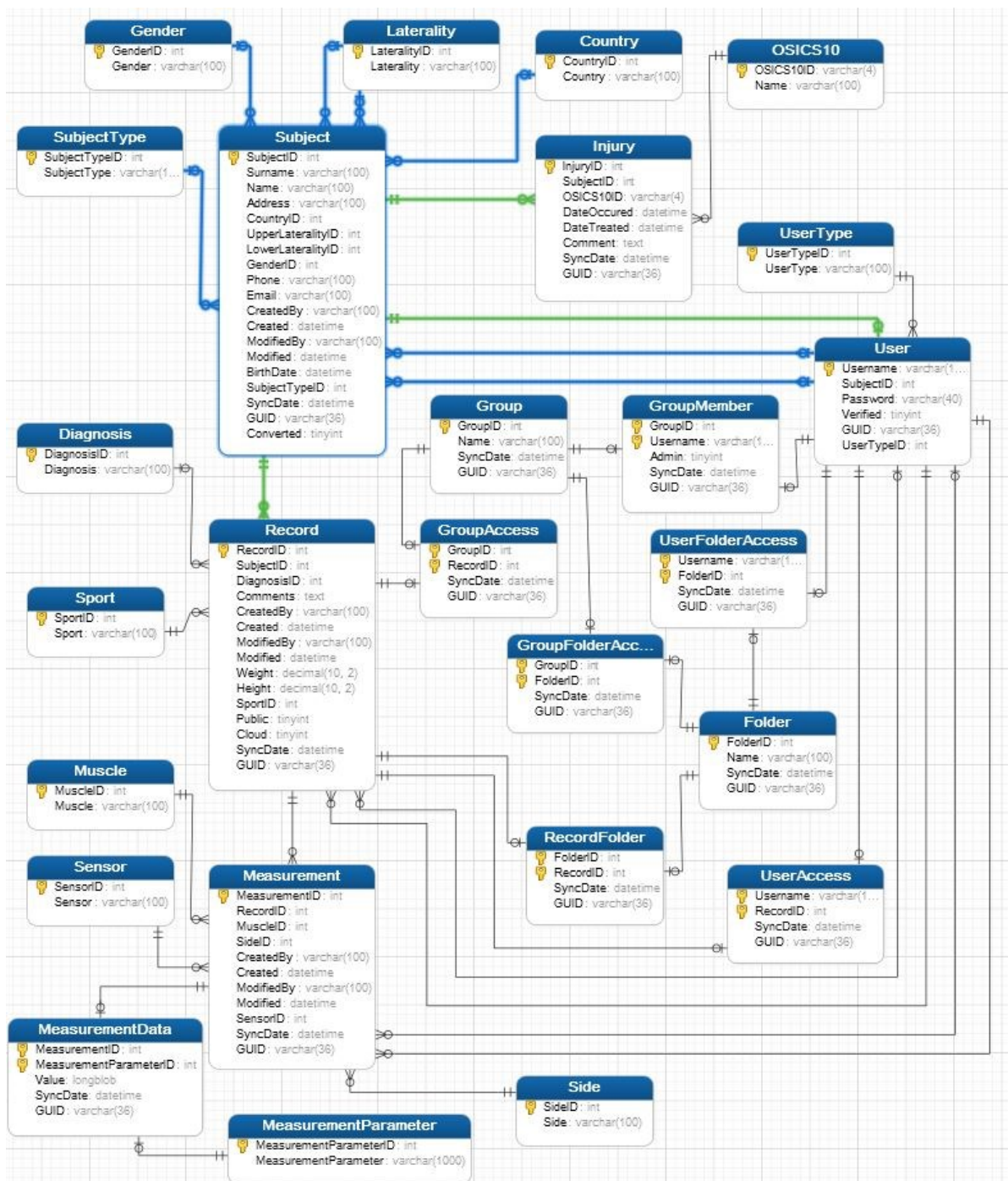
Arhitekturo celotnega sistema prikazuje Slika 4.1. Spletna aplikacija se nahaja na spletnem strežniku Apache, tako da je z uporabnikovega vidika arhitekturno neodvisna. Do nje dostopa preko spletnega brskalnika. Aplikacija črpa podatke iz podatkovne baze, ko se nahaja v oblaku. Za uporabo spletne aplikacije in pregled podatkov potrebuje vsak uporabnik aplikacije uporabniško ime ter geslo.



Slika 4.1: Arhitektura celotnega sistema.

## 4.2 Podatkovna baza

Aplikacija je razvita za že obstoječo podatkovno bazo, ki se nahaja v oblaku (Amazon RDS). Tip baze je MySQL. Sestavlja jo 26 tabel (Slika 4.2). Poleg tabel s podatki o uporabnikih aplikacije, spadajo tabele še v štiri glavne skupine: podatki o merjencih, podatki o meritvah, podatki o zapisih ter podatki o pravicah za dostopanje.



Slika 4.2: Entitetni diagram.



Glavne entitete so sledeče:

- **User**

Entiteta uporabnik (angl. *user*) predstavlja uporabnika aplikacije.

- **Subject**

To je entiteta merjenec (angl. *subject*) oziroma oseba, katere podatke o mišičnih lastnostih držimo v bazi.

- **Record**

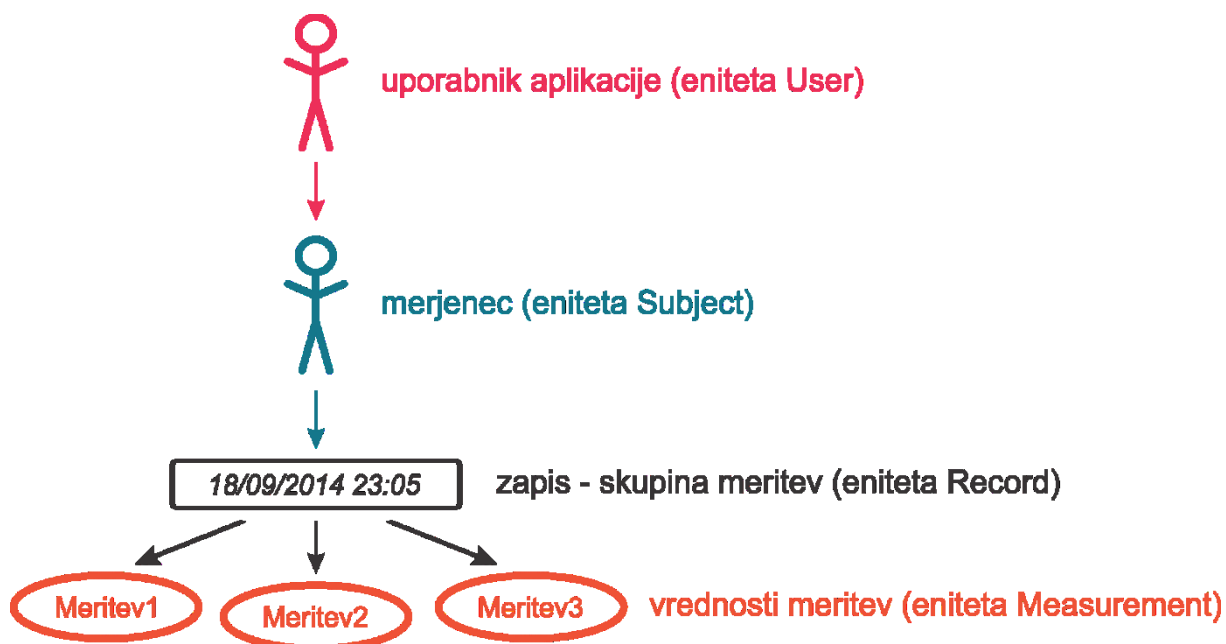
Entiteta zapis (angl. *record*) predstavlja skupino večih meritev mišičnih lastnosti.

- **Measurement**

Entiteta meritev (angl. *measurement*) drži vrednosti posamezne meritve.

- **Entitete za pravice dostopov**

Predstavljajo organizacijo dostopanja do podatkov.



Slika 4.3: Shema organizacije podatkov.

### 4.3 Dostopanje do podatkov in njihova interpretacija

Če želimo podatke analizirati in prikazovati v predstavitvenih oblikah, kot so grafi, jih je potrebno zajeti iz baze podatkov, jih obdelati ter jih nato uporabniku spletne aplikacije ustrezno predstaviti.

Vsaka meritvev ima v bazi zabeležene tri različne zapise:

- dejanski rezultati meritve, ki jih vrača senzor za merjenje mišice;
- električni tok, ki je bil uporabljen pri merjenju mišice;
- karakteristične točke meritve.

V našem primeru nas zanima zadnji zapis – karakteristične točke meritve (Td, Tc, Ts, Tr, Dm), katere so podrobneje opisane v 2. poglavju. V podatkovni bazi so vse vrednosti shranjene v enem polju, tipa »longblob«, v katerem so vrednosti med seboj ločene z navpično ločnico »|«.

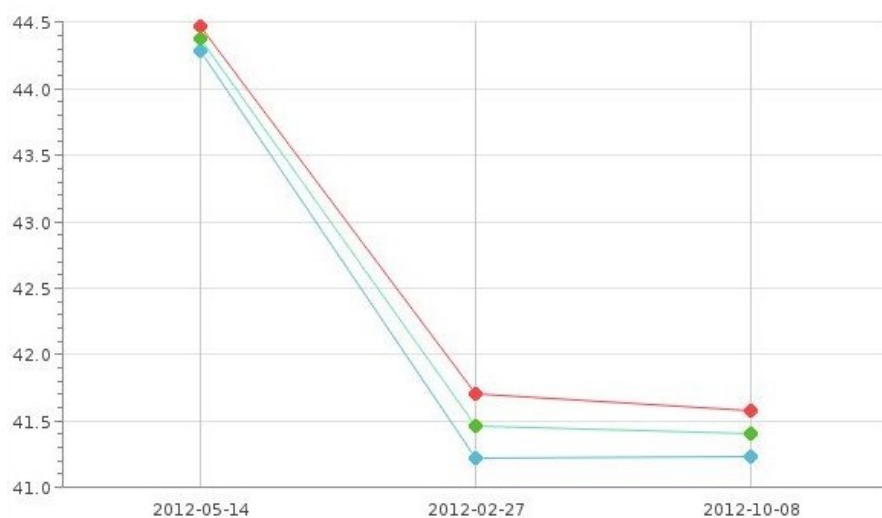
Ker omogoča ORM sistem Eloquent preprost način zajema podatkov iz podatkovne baze in delo z njimi, jih obdelujemo šele na aplikacijskem nivoju. Na tabele iz podatkovne baze s pomočjo modelov (angl. *models*), na podatke gledamo kot na objekte, ki držijo vrednosti iz podatkovne baze. Podatke preberemo in obdelamo (poiščemo minimalne in maksimalne vrednosti, izračunamo povprečja, razdelimo jih v skupine glede na meritve leve in desne strani telesa ...) in jih pripravimo v podatkovne nize, ki služijo kot osnova za prikaz grafov. Za izrisovanje smo uporabili knjižnico za risanje grafov JpGraph [16], ki temelji na jeziku PHP in jo zato enostavno uporabimo v naši spletni aplikaciji.

### 4.4 Prikazovanje podatkov

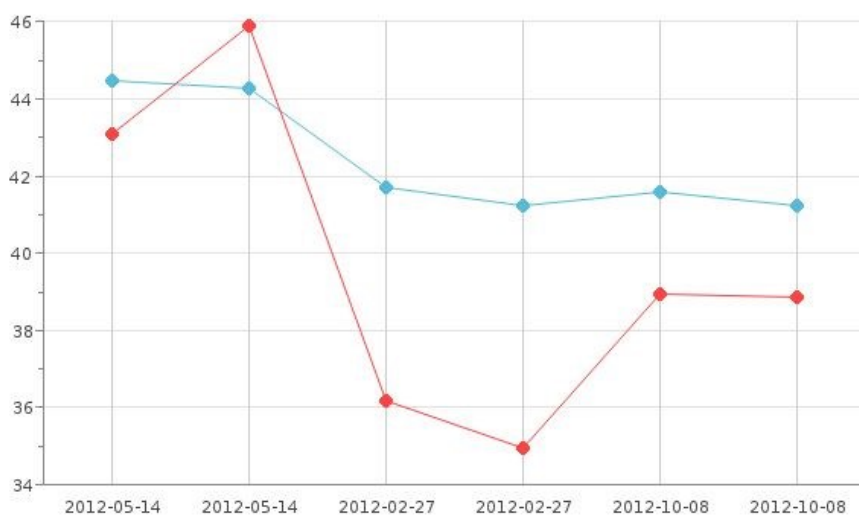
Aplikacija nas postopoma vodi skozi postopek izbiranja parametrov, ki so potrebni za zajem podatkov, ki jih želimo prikazati v grafih. Poleg izbranih parametrov se iz podatkovne baze prepišejo še drugi podatki o izbranem merjencu, zapisu ali meritvah (npr. vrsta športa, za katerega so bile izvedene meritve mišic na merjencu). Aplikacija prikazuje končne podatke v treh različnih oblikah grafov:

- linearni graf spreminjanja vrednosti karakterističnih točk skozi čas z minimalno, maksimalno in izračunano povprečno vrednostjo prikazanih meritev v zajetih zapisih (Slika 4.4),

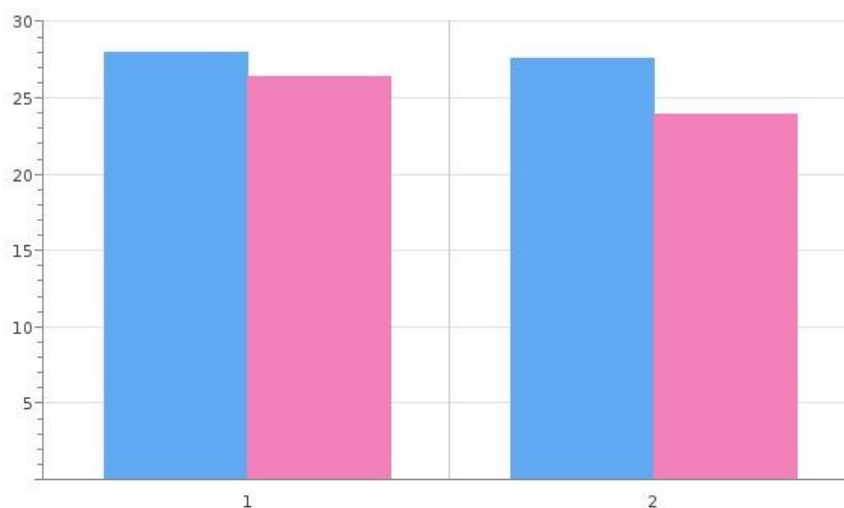
- linearni graf primerjave leve in desne strani spreminjanja karakterističnih točk skozi čas (Slika 4.5),
- stolpčni graf vrednosti meritev za izbrano mišico v določenem zapisu (Slika 4.6).



Slika 4.4: Graf spreminjanja vrednosti karakterističnih točk skozi čas.



Slika 4.5: Graf primerjave vrednosti meritev leve in desne strani.



Slika 4.6: Graf vrednosti meritev za izbrano mišico v zapisu.

## 4.5 Uporaba aplikacije

Za uporabniški vmesnik je pomembno, da je čim bolj enostaven, pregleden in preprost za uporabo. Ker gre za spletni uporabniški vmesnik, že v osnovi velja nenapisano pravilo, da morajo biti vse funkcionalnosti in podatki oddaljeni le nekaj klikov stran (običajno največ trije kliki). To je dandanes vedno bolj pomembno, saj so namesto osebne računalnika ali prenosnega računalnika vedno pogosteje v uporabi tablični računalniki in mobilni telefoni. Zato se vedno več spletnih strani izogiba gnezdenju podstrani in namesto tega prikazuje vsebino razvrščeno zaporedno vzdolž spletne strani. Tako je vsebina lažje dostopna z drsenjem po ekranu na dotik, vseeno pa ostaja stran pregledna in uporabna za pregledovanje na klasičnem osebnem računalniku z uporabo miške.

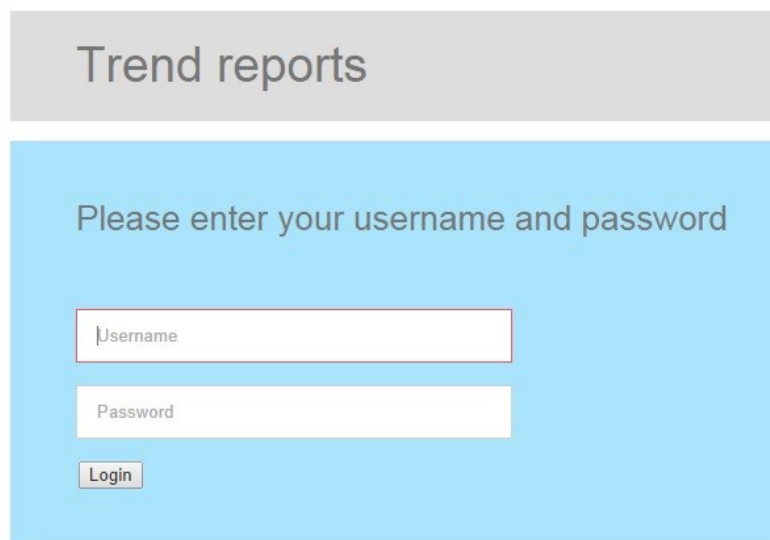
Vsi podatki, ki jih aplikacija uporablja, se nahajajo v podatkovni bazi, v več tabelah. Za dostopanje do teh podatkov potrebujemo vmesnik, ki je v našem primeru spletna aplikacija. Nekateri podatki so shranjeni v oblikah, ki jih brez predhodne obdelave ne moremo brati direktno, zato spletna aplikacija poskrbi za njihovo obdelavo pred prikazom.

Glavne funkcionalnosti aplikacije z uporabniškega vidika so:

- prikaz spreminjanja karakterističnih točk skozi čas v obliki grafikonov,
- pregled vrednosti posameznih meritev,
- primerjava vrednosti meritev glede na levo in desno stran.

### 4.5.1 Prijavno okno

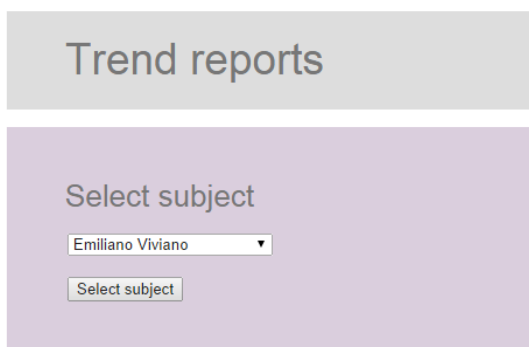
Preden lahko spletno aplikacijo uporabljamo se moramo prijaviti. Prijavno okno je prva stran, ki jo vidimo pri dostopanju do spletne aplikacije (Slika 4.7). Vnesti moramo svoje uporabniško ime in geslo, sicer do aplikacije ne moremo dostopati. Ko vnesemo podatke za prijavo, kliknemo na gumb »Login«. Ko se uspešno prijavimo v aplikacijo, se nam prikaže vmesnik za izbiro merjenja, ki ga želimo pregledovati.



Slika 4.7: Prijavno okno spletne aplikacije.

### 4.5.2 Izbira merjenja

Aplikacija nam takoj v naslednjem koraku ponudi izbiro merjenja, za katerega želimo pregledovati podatke (Slika 4.8). Vse merjenja nam ponudi v obliki spustnega seznama, zato moramo poznati merjenčevo ime ter priimek. Ko na seznamu izberemo merjenja, svojo izbiro potrdimo s klikom na gumb »Select subject«.



Slika 4.8: Okno za izbiro merjenja.

### 4.5.3 Izbira funkcionalnosti

Ko smo izbrali merjenca, za katerega želimo pregledovati podatke, se nam prikaže glavno okno aplikacije, ki nam ponuja več možnosti za pregled podatkov o izbranem merjencu. Okno je razdeljeno na tri glavne dele, kot prikazuje Slika 4.9. Prvi del vsebuje podatke o izbranem merjencu, ter gumb »Change subject«, ki nas v primeru, da želimo spremeniti merjenca, vrne na prejšnji korak. Drugi, sredinski del, je namenjen za prikaz spreminjanja karakterističnih točk skozi čas ter primerjavo karakterističnih točk izbrane mišice glede na levo in desno stran mišice. Zadnji del pa nam omogoča pregled vrednosti točno določenega zapisa ter izbrane meritve.

Subject: **Emiliano Viviano**

Gender: **Male**

[Change subject](#)

### TREND REPORT

Select muscle

Select muscle: **1 - BB Biceps Brachii**

Compare left and right side for selected muscle ☐

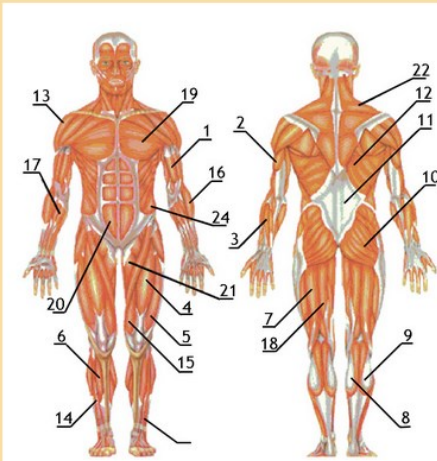
Select muscle side: **Right**

Optional (empty means not limited with dates):

Date from:

Date to:

[Show report](#)



1	Biceps Brachii
2	Triceps Brachii
3	Brachioradialis
4	Rectus Femoris
5	Vastus Lateralis
6	Tibialis Anterior
7	Biceps Femoris
8	Gastrocnemius Medialis
9	Gastrocnemius Lateralis
10	Glatius Maximus
11	Erector Spinae
12	Latissimus Dorsi
13	Deltoides
14	Soleus
15	Vastus Medialis
16	Extensor Digitorum
17	Flexor Digitorum
18	Semitendinosus
19	Pectoralis Major
20	Rectus Abdominis
21	Adductor Langus
22	Trapezius
23	Peroneus Brevis
24	Obliques Externus

### VIEW MEASUREMENT DATA

Select record

**2012-05-14 14:30:06**

Select muscle

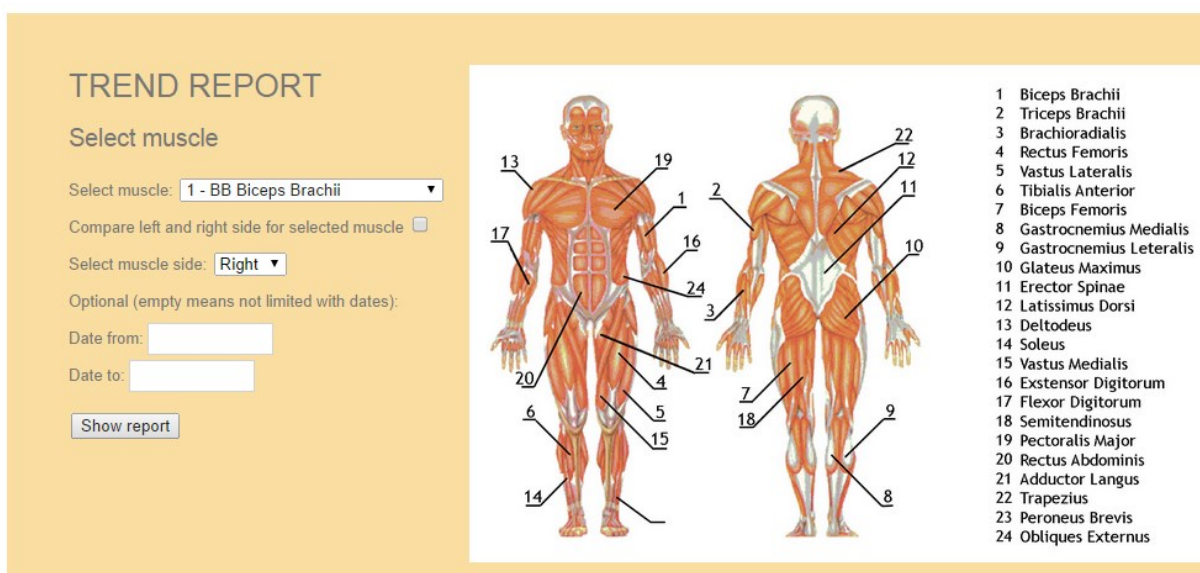
**1 - BB Biceps Brachii**

[Show report](#)

Slika 4.9: Glavno okno aplikacije.

#### 4.5.4 Prikaz spreminjanja karakterističnih točk skozi čas

Če nas zanima spreminjanje karakterističnih točk skozi čas, izberemo razdelek z naslovom »Trend report«, kot prikazuje Slika 4.10. S prvega spustnega seznama v razdelku moramo izbrati mišico, ki jo želimo obravnavati. Če postavimo kljukico pri »Compare left and right side for selected muscle«, bomo v nadaljevanju primerjali levo in desno stran telesa za izbrano mišico, sicer pa nam bo vmesnik omogočil dodaten spustni seznam, s katerega izberemo, katera stran telesa nas zanima (leva ali desna). V tem primeru nam bo aplikacija prikazala grafe za karakteristične točke samo za izbrano mišico, za izbrano levo ali desno stran telesa. Opcijsko lahko nastavimo še datumsko območje, znotraj katerega bi radi omejili pregledovanje podatkov.



**TREND REPORT**

Select muscle

Select muscle: 1 - BB Biceps Brachii

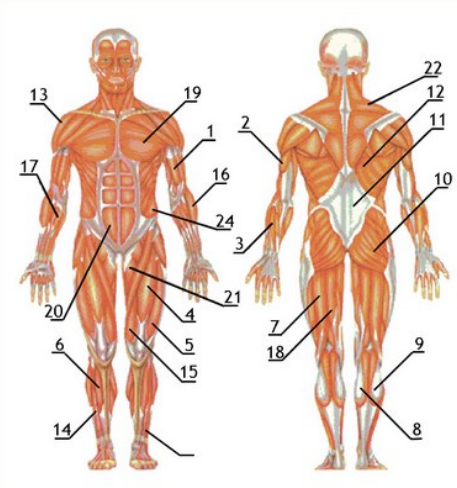
Compare left and right side for selected muscle ☐

Select muscle side: Right

Optional (empty means not limited with dates):

Date from:

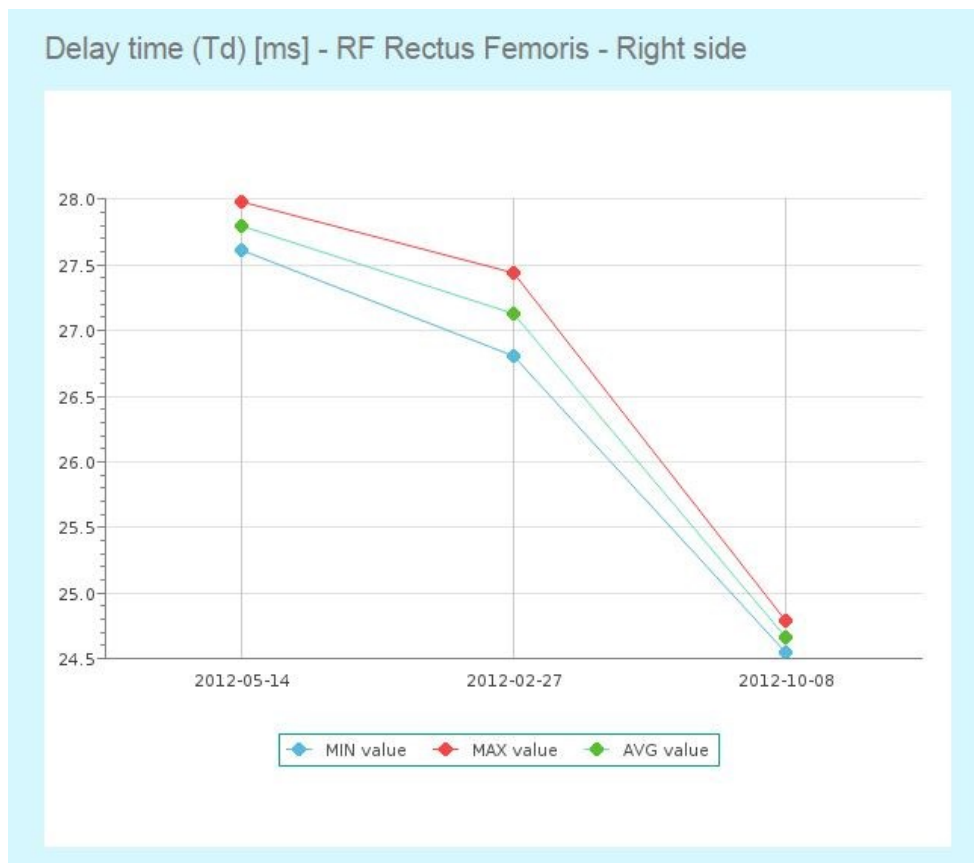
Date to:

	<ul style="list-style-type: none"> <li>1 Biceps Brachii</li> <li>2 Triceps Brachii</li> <li>3 Brachioradialis</li> <li>4 Rectus Femoris</li> <li>5 Vastus Lateralis</li> <li>6 Tibialis Anterior</li> <li>7 Biceps Femoris</li> <li>8 Gastrocnemius Medialis</li> <li>9 Gastrocnemius Lateralis</li> <li>10 Gluteus Maximus</li> <li>11 Erector Spinae</li> <li>12 Latissimus Dorsi</li> <li>13 Deltodeus</li> <li>14 Soleus</li> <li>15 Vastus Medialis</li> <li>16 Extensor Digitorum</li> <li>17 Flexor Digitorum</li> <li>18 Semitendinosus</li> <li>19 Pectoralis Major</li> <li>20 Rectus Abdominis</li> <li>21 Adductor Langus</li> <li>22 Trapezius</li> <li>23 Peroneus Brevis</li> <li>24 Obliques Externus</li> </ul>
---	--

Slika 4.10: Sredinski del aplikacije za vnos podatkov.

Ko z gumbom »Show report« potrdimo vnesene podatke, nam aplikacija prikaže ustrezne podatke v obliki petih grafikonov, za vsako karakteristično točko posebej. Če zelenih podatkov o meritvah ni, nas aplikacija na to opozori.

Graf, ki ga prikazuje Slika 4.11, predstavlja karakteristično točko »čas zakasnitve« (angl. *delay time*) za mišico *Rectus Femoris* (sprednja stegenska mišica) na desni strani.



Slika 4.11: Graf spreminjanja časa zakasnitve sprednje stegenske mišice na desni strani.

Na X osi prikazanega grafikona so predstavljeni vsi zapisi, ki so zajeti v izbranem datumskem obdobju, na Y osi pa so predstavljene vrednosti meritev v milisekundah za izbrano mišico, za karakteristično točko, ki jo predstavlja grafikon. Prikazane so tri krivulje:

- minimalna vrednost meritev v zapisu (modra krivulja),
- maksimalna vrednost meritev v zapisu (rdeča krivulja),
- povprečna vrednost meritev v zapisu (zelena krivulja).

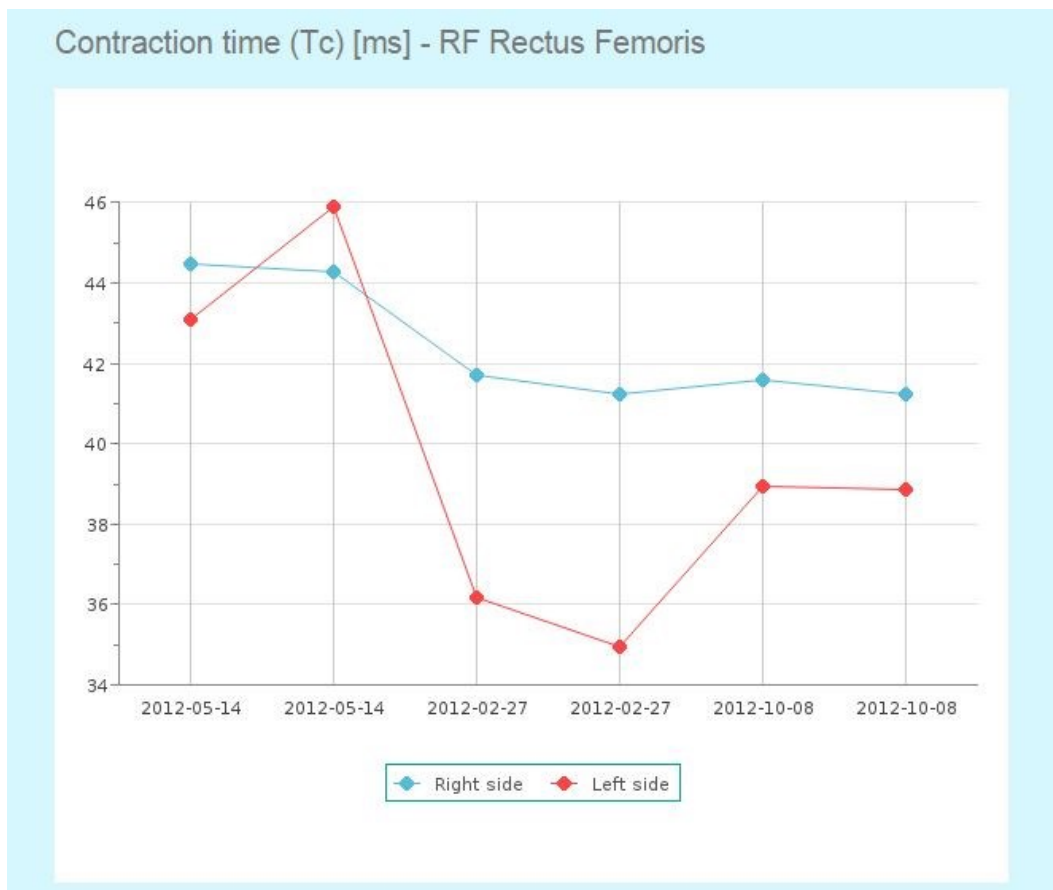
Iz krivulje lahko razberemo natančnost meritve za posamezen zapis – manj, kot se razlikujeta maksimalna in minimalna vrednost, bolj so meritve natančne. Iz grafa, ki ga prikazuje Slika 4.11 je razvidno, da je razlika med maksimalno in minimalno vrednostjo občutno manjša pri zapisu »2012-10-08«, v katerem imajo vse meritve občutno manjše zakasnitvene čase kot pri ostalih dveh zapisih. V istem primeru smo opazili, da velja obratno za čas krčenja (angl. *contraction time*), kjer je bila pri višjih časih krčenja manjša razlika med minimalno in maksimalno vrednostjo meritev.



#### 4.5.5 Prikaz spreminjanja karakterističnih točk skozi čas

Če v razdelku »Trend report«, ki ga prikazuje Slika 4.10, postavimo kljukico »Compare left and right side for selected muscle«, nam aplikacija spusti seznam za izbiro strani telesa, na kateri se nahaja mišica, onemogoči, saj to pomeni, da želimo primerjati levo in desno stran telesa za isto mišico, ki smo jo izbrali s spustnega seznama vseh mišic. Ko kliknemo na gumb »Show report«, nam aplikacija izriše pet grafov, za vsako karakteristično točko, ki prikazujejo spreminjanje vrednosti leve in desne mišice.

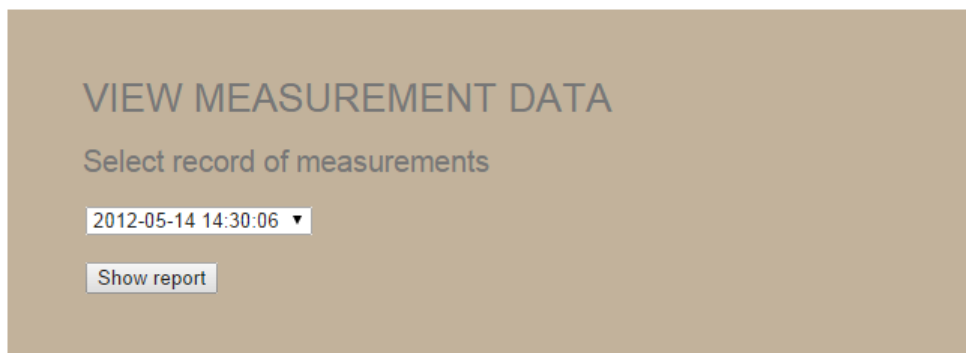
Graf, ki ga prikazuje Slika 4.12, predstavlja vrednosti meritev časa krčenja obeh sprednjih stegenskih mišic. Na X osi so navedene vse zajete meritve, na Y osi pa so časi krčenja mišice v milisekundah. Modra krivulja predstavlja desno, rdeča krivulja pa levo stran telesa. Iz grafa se lepo vidi, da je čas krčenja na desni strani veliko bolj konstanten in uravnotežen kot na levi strani, kjer se mišica odziva zelo različno od meritve do meritve.



Slika 4.12: Graf spreminjanja časa krčenja sprednje stegenske mišice.

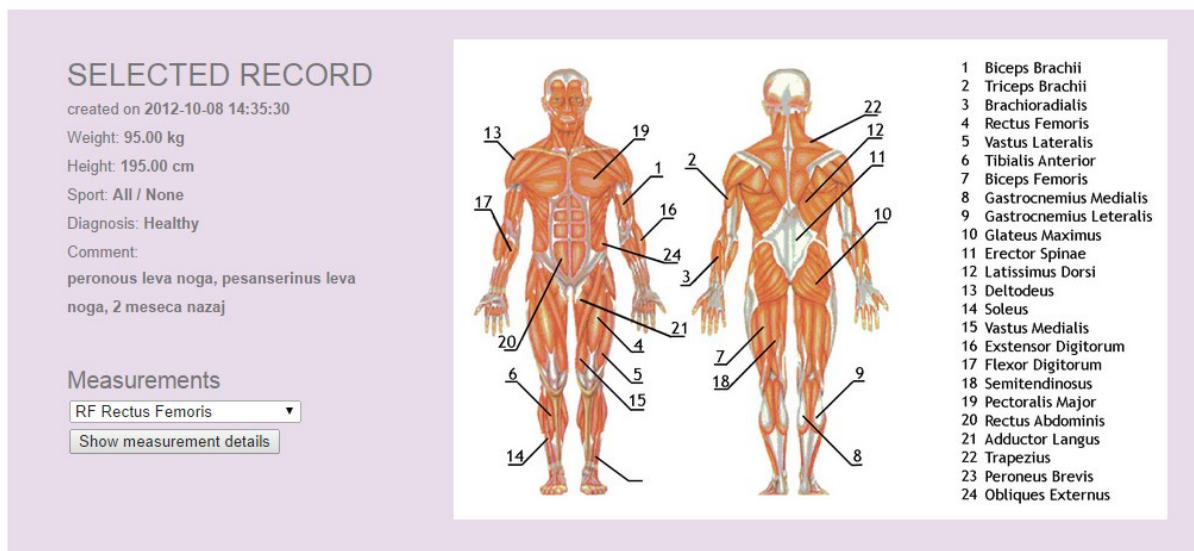
### 4.5.6 Pregled vrednosti posameznih meritev

Zadnji razdelek glavnega okna (Slika 4.13), nam omogoča pregled podatkov za posamezni zapis meritev ter izris grafov vrednosti meritev v izbranem zapisu. S spustnega seznama izberemo zapis, ki si ga želimo ogledati podrobneje in kliknemo na gumb »Show report«.



Slika 4.13: Vnos podatkov za pregledovanje točno določene meritve.

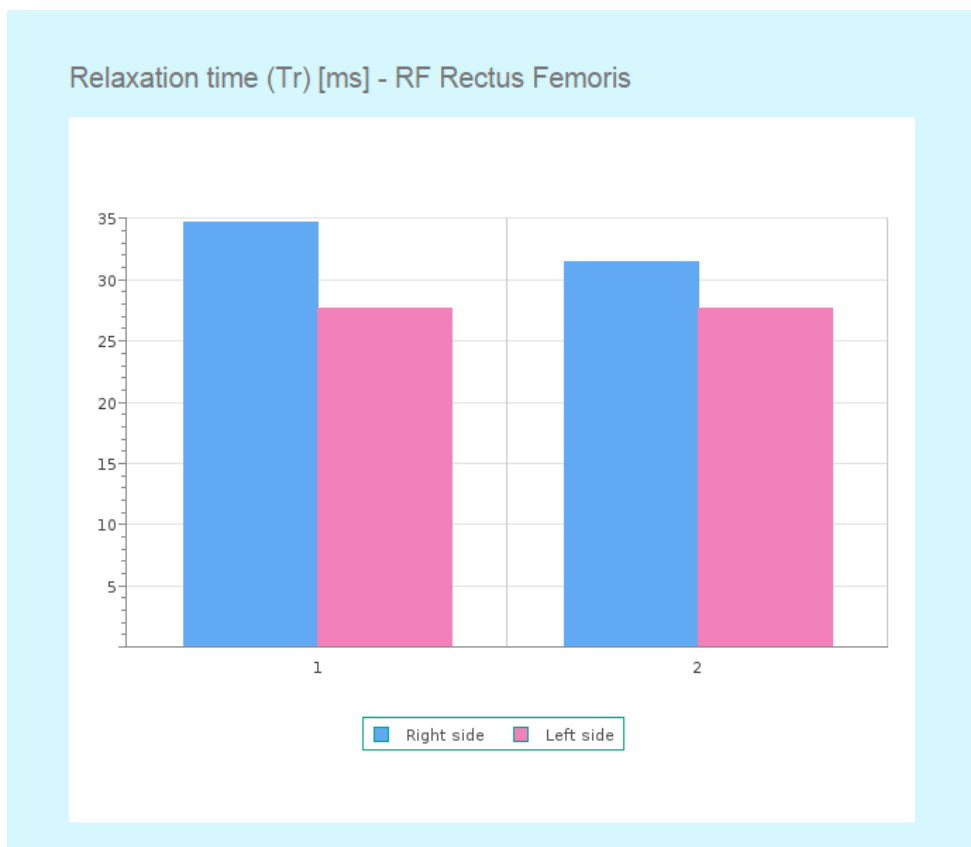
Prikaže se okno s podrobnejšimi informacijami o izbranem zapisu meritev in merjencu ter spustnim seznamom vseh mišic, za katere so bile izvedene meritve v sklopu izbranega zapisa (Slika 4.14).



Slika 4.14: Podrobnejše informacije o izbranem zapisu.

Če si želimo ogledati graf vrednosti meritev za eno od mišic v zapisu, jo poiščemo na spustnem seznamu mišic in kliknemo gumb »Show report«. Prikaže se nam pet stolpcnih grafov, za vsako karakteristično točko posebej.

Graf, ki ga prikazuje Slika 4.15, predstavlja čas sprostitve leve in desne stegenske mišice za izbrani zapis. Na X osi so navedene vse meritve izbrane mišice iz izbranega zapisa. Na Y osi pa so vrednosti za čas sprostitve mišic v milisekundah. Iz grafa se lepo vidi, da se v obeh meritvah hitreje sprosti mišica na levi strani.



Slika 4.15: Čas sprostitve stegenskih mišic v izbranem zapisu.



## Poglavje 5      Zaključek

V diplomski nalogi smo se ukvarjali s prikazovanjem in analizo tenziomiografskih meritev. Razvili smo aplikacijo, glede na obstoječe podatke o meritvah, ki nam v grobi obliki ne povedo veliko, še manj pa jih lahko primerjamo med seboj. Za čim bolj nazoren prikaz smo uporabili več različnih vrst grafov, ki nam numerične podatke nazorno prikažejo v grafični obliki na načine, da se iz njih da razbrati določene lastnosti, kot na primer primerjava mišic leve in desne strani telesa ali napredek določene lastnosti mišice skozi čas. V pregledu uporabljenih metod smo se spoznali z MVC arhitekturo. Pri razvoju aplikacije smo uporabili ogrodje Laravel, ki uporablja to arhitekturo in nam omogočilo razvoj zelo modularne rešitve. Skozi razvoj rešitve smo spoznali večinski del ogrodja.

Največ težav je prineslo nepoznavanje ogrodja, saj smo ga uporabljali prvič. Ogrodje na eni strani dostopa do podatkovne baze, na drugi strani pa izrisuje grafe uporabniku. Prav tako pa procesira vse vmesne korake, kar nas je prisililo, da smo preko vseh nivojev ogrodja podrobneje spoznali.

Rešitev smo uspešno razvili do te mere, da spletna aplikacija izpolnjuje cilje, ki smo si jih zadali na začetku. Nastal je pregleden in uporaben uporabniški vmesnik, ki omogoča prikaz podatkov v obliki grafov, ki so primerni za nadaljnjo analizo. Hitrost spletnega vmesnika je zadovoljiva, najožje grlo predstavlja dostopanje do podatkov, ki se nahajajo v oblaku, do katerega mora aplikacija dostopati na vsakem izvedenem koraku.

Naša rešitev ima še veliko možnosti izboljšav in razširitev. Kot možne razširitve so v prvi vrsti še drugi na tenziomiografske meritve ter njihova prezentacija končnemu uporabniku. Sam uporabniški vmesnik je možno še funkcionalno izboljšati (npr. mišico bi lahko izbrali s slike človeškega telesa in ne s spustnega seznama). Z nekaj izboljšavami bi lahko spletno aplikacijo še bolj prilagodili za uporabo na mobilnih telefonih.



## Literatura

- [1] Patent EP1424938A1 [Online]. Dosegljivo: <http://www.google.com/patents/EP1424938A1?cl=en>.
- [2] TMG – BMC d.o.o. Tenziomiografija, Celovita diagnostika mišic [Online]. Dosegljivo: <http://www.bravo1.si/wp-content/uploads/2012/10/TMGeBrochureSLO.pdf.pdf>.
- [3] E. Rey, C. Lago-Peñas, J. Lago-Ballesteros. (2012). Tensiomyography of selected lower-limb muscles in professional soccer players, *Journal of Electromyography and Kinesiology*. [Online]. 22 (2012), str. 866–872. Dosegljivo: [http://www.jelectromyographykinesiology.com/article/S1050-6411\(12\)00113-7/abstract](http://www.jelectromyographykinesiology.com/article/S1050-6411(12)00113-7/abstract).
- [4] Navicat Premium [Online]. Dosegljivo: <http://www.navicat.com/products/navicat-premium>. [Dostopano 23.7.2014].
- [5] Tomcat Apache [Online]. Dosegljivo: [http://sl.wikipedia.org/wiki/Spletni\\_stre%C5%BEnik\\_Apache](http://sl.wikipedia.org/wiki/Spletni_stre%C5%BEnik_Apache). [Dostopano 23.7.2014].
- [6] Sublime text [Online]. Dosegljivo: <http://www.sublimetext.com>. [Dostopano 23.7.2014].
- [7] Laravel framework. Dosegljivo: <http://en.wikipedia.org/wiki/Laravel>. [Dostopano 24.7.2014].
- [8] Understanding Model-View-Controller [Online]. Dosegljivo: <http://blog.codinghorror.com/understanding-model-view-controller>. [Dostopano 26.7.2014].

- [9] Trinivojska arhitektura sistemov [Online]. Dosegljivo: <http://www.ditea.si/sl/tehnologije/>. [Dostopano 17.8.2014].
- [10] HTML [Online]. Dosegljivo: <http://sl.wikipedia.org/wiki/HTML>. [Dostopano 27.7.2014].
- [11] CSS [Online]. Dosegljivo: <http://sl.wikipedia.org/wiki/CSS>. [Dostopano 28.7.2014].
- [12] JavaScript in HTML [Online]. Dosegljivo: <http://zaversnik.fmf.uni-lj.si/gradiva/javascript/uvod/script.php>. [Dostopano 1.8.2014].
- [13] Uvod v jQuery [Online]. Dosegljivo: <http://ussi.feri.uni-mb.si/index.php/vaje/vaja08/123-uvod-v-jquery>. [Dostopano 1.8.2014].
- [14] SQL [Online]. Dosegljivo: <http://sl.wikipedia.org/wiki/SQL>. [Dostopano 3.8.2014].
- [15] Kaj je PHP?. Dosegljivo: <http://www.presentia.si/baza-znanja-helpdesk/2008/kaj-je-php>. [Dostopano 6.8.2014].
- [16] JpGraph [Online]. Dosegljivo: <http://jpgraph.net>. [Dostopano 15.8.2014].